What is claimed is:

1. A method for distribution of a task, by a host computer, to a device that comprises an address and a processor having an idle state, the method comprising the steps of:

formatting the task and execution instructions in a packet;

identifying the packet for processing by the processor of the device during a period when the processor is normally in the idle state;

transmitting the packet to the device for generation of a result file by the processor in response to the execution instructions; and

receiving the results file.

2. The method of claim 1 wherein the processor runs an alternate personality of a plurality of personalities and the packet is processed by the alternate personality.

3. The method of claim 2 wherein the plurality of personalities comprises one or more of a POSTSCRIPT or PCL personality.

4. The method of claim 1 wherein the address is a network address.

5. The method of claim 4 wherein the network address is an Internet protocol address.

6 The method of claim 3 wherein the network address is an Ethernet address.

7. The method of claim 1 and further including the step of the host computer transmitting an executable file to the device for use by the processor in order to process the task.

8. The method of claim 1 wherein the device identifies the task as an idle state task in response to a port of the device over which the packet is received.

9. A method for distribution of a task, by a host computer, to a device that comprises an address, an operating system that performs a plurality of personalities, and a processor that executes the personalities and operating system, the method comprising the steps of:

identifying the task at the host computer for processing by the device;

formulating the task into executable instructions;

wrapping task execution instructions and the task in a packet;

labeling the packet for processing by a first personality of the plurality of personalities;

addressing the packet with the address;

transmitting the packet to the device for generation by the first personality of results in response to the task and the task execution instructions; and

receiving the results from the device.

10. The method of claim 9 wherein the plurality of personalities comprise a POSTSCRIPT personality and a PCL personality and the first personality comprises an idle CPU personality.

11. The method of claim 9 wherein the step of formulating the task into an executable state comprises formulating the task into JAVA code.

12. The method of claim 9 wherein the task execution instructions comprise an executable file for execution of the task.

13. The method of claim 12 wherein the task comprises a data file that is used by the executable file to generate the results.

14. A method for distribution of a task, by a host computer, to a printer that comprises an operating system that includes a JAVA interpretation process and a processor that executes the JAVA interpretation process and operating system, the printer having an idle state during which printing is not performed, the method comprising the steps of:

identifying the task at the host computer for processing by the printer;

formulating the task into an executable form comprising JAVA code;

wrapping task execution instructions and the JAVA code in a packet;

labeling the packet for processing by the JAVA interpretation process;

transmitting the packet to the printer for generation of results by the JAVA interpretation process in response to the JAVA code and the task execution instructions; and

the host computer receiving the results from the printer.

15. A printer apparatus coupled to a host computer, the apparatus comprising:

a processor having an idle state during which printing is not performed;

an operating system, executed by the processor, for performing various personalities of the printer apparatus;

means for receiving a packet from the host computer, the packet comprising a task to be performed by the processor during the idle state;

means for interpreting the task and generating task results; and

means for transmitting the task results back to the host computer.

16. The printer apparatus of claim 15 wherein the task is written in JAVA code and the means for interpreting the task comprises a JAVA Virtual Machine.

17. The printer apparatus of claim 15 wherein the various personalities of the printer apparatus comprise a POSTSCRIPT personality, a PCL personality, and an idle CPU task personality.

18. A computer system for minimizing processing time for large processing job requests, including a computer in communication with at least one remote peripheral device having a processor, memory, and an operating system, the system comprising:

means for parsing tasks from the large processing job request for processing by the at least one remote peripheral device;

means for generating a task comprising data and execution instructions;

means for wrapping the task with a functionality label to form a packet;

means for transmitting the packet to the at least one remote peripheral device for processing by the at least one remote peripheral device to generate task results; and

means for receiving the task results from the at least one remote peripheral device.

19. The system of claim 18 and further comprising:

means for receiving the packet at the at least one remote peripheral device;

means for determining a necessary functionality for processing the task from the wrapper label;

means for unwrapping the packet;

means for processing the task with the necessary functionality, according to the execution instructions, and generating the task results;

means for capturing the task results; and

means for addressing the task results for return to a transmitting computer.

20. The computer system of claim 19 wherein the necessary functionality is a JAVA Virtual Machine.

21. The computer system of claim 19 wherein the at least one remote peripheral device is one of a printer, a scanner, gaming systems, and a personal digital assistant.

22. The computer system of claim 19 and further including means for storing the task in memory of the at least one remote peripheral device.

23. A computer system for minimizing processing time for large processing job requests, the system comprising:

a computer having a processing unit and memory that stores programming commands that, when read by the processing unit, causes the processing unit to function to: parse a task from the large processing job request, and wrap the task and instructions for processing the task with a functionality label to form a packet; and

at least one remote peripheral device having a processing unit and memory that stores programming commands that, when read by the peripheral processing unit, causes the peripheral processing unit to function to: receive a packet from the computer, wherein the packet includes the functionality label, the task and the instructions for processing the task, determine a necessary functionality for processing the task in response to the wrapper label, unwrapping the packet, and processing the task with the necessary functionality to generate task results.

24. The computer system of claim 23 wherein the necessary functionality is a JAVA Virtual Machine.

25. The computer system of claim 23 wherein the task comprises at least one data file.

26. The computer system of claim 23 wherein the task comprises at least one executable file.

27. The computer system of claim 24 wherein the task is written by the processor in a code that is interpreted by the JAVA Virtual Machine.